

**Amendments to the Specification:**

**Please replace Paragraph [0008] with the following amended paragraph:**

5 [0006] Referring to FIG. 2, an MP3 file contains a number of frames. Each frame is a unit  
of data that is enough for decoding and generating a certain amount of audio samples.  
Take frame 7 for example. The frame 7 includes a header 71, an error check 72, a sample  
data 73, and an external data 74. The header 71 is a 32-bit stream including a 12-bit  
synchronization word. In addition, the header 71 has 2 bits for sampling frequency, 4 bits  
10 for bitrate, and other information. The error check 72 is an optional 16-bit stream for error  
detection. Sample data 73 includes necessary information for decoding audio samples of  
this frame. External data 74 includes data other than audio information. Sample data 73  
includes main\_data\_begin 731, audio\_data 732, and main\_data 733. The  
main\_data\_begin 731 will be described later. The audio\_data 732 contains parametric  
15 information about the decoding algorithm and main\_data 733 contains data of the  
encoded audio samples. The size of audio\_data 732 and main\_data 733 may vary in  
different frames. Although the contents and size of frames may vary from frame to frame,  
each of the frames, such as frames 1, 4, 5, 6, 7, and 8 shown in Fig.2, comprise the same  
structure, for example each of the frames 1, 4, 5, 6, and 8 comprise a main\_data field  
20 corresponding to the main\_data 733 of frame 7 and a main\_data\_begin field  
corresponding to the main\_data\_begin 731 of Fig.7.

**Please replace Paragraph [0008] with the following amended paragraph:**

[0008] For example, the size of the main\_data 733 of a certain frame is 100 bytes and the required size of the audio samples is 203 bytes. The 103 bytes of overflow will be allocated to a main\_data of prior frames, such as a main\_data [[633]] of a frame 6. If the size of the main\_data [[633]] of frame 6 is still not enough for storage of the 103 bytes, a  
5 main\_data [[533]] of a frame 5 will be used. The main\_data\_begin 731 will have a value of 103 indicating the size of the overflow is 103 bytes. If the size of the main\_data [[633]] of frame 6 is only 70 bytes, then there will still be another 33 bytes of overflow. The 33 bytes of overflow will again be allocated to the main\_data [[533]] of the frame 5. To avoid an error, the MP3 standard requests that the main\_data\_begin of the first frame of  
10 an MP3 file, which is frame 1 in FIG. 2, must be zero.

**Please replace Paragraph [0015] with the following amended paragraph:**

[0015] FIG. 2 is a block diagram of ~~a traditional~~ an MP3 ~~player~~ file.

**Please replace Paragraph [0017] with the following amended paragraph:**

[0017] The present invention allows the correct playing of an MP3 file from any frame in  
15 the MP3 file and in either direction while eliminating the "bang" noise. The present method only needs to be invoked when beginning to play an MP3 file and works by properly selecting a first frame for which all data associated with the frame to be decoded has been read. Decoding the main\_data of the frame is preferably processed using the Huffman decoding method. Such MP3 file decoding methods are well known in the art  
20 and will not be discussed here.

**Please replace Paragraph [0020] with the following amended paragraph:**

[0020] The next step 902 involves searching the bit stream for a first occurrence of a 12 bit synchronization word. When the synchronization word 711 has been located, the header 71 of the frame 7 is decoded providing bitrate and other miscellaneous

information (step 903). The header 71 includes a bit signifying whether or not error checking is to be used within the frame. If error checking is used in this MP3 file, the error check 72 is read next (step 904), but, as with any MP3 player 10, using the error check 72 is merely a design choice and the present invention is not limited to the reading or not reading of this field. Next, the sample data 73 is decoded, the main\_data\_begin 731 is read (step 905). Then the size of the main\_data 733 is calculated (step 906). ~~905~~.

**Please replace Paragraph [0023] with the following amended paragraph:**

[0023] While the parser 101 parsed the series of frames to the decoder 103, the parser 101 will also notify the decoder 103 whether to decode the audio samples from the beginning of an MP3 file, or from the middle of an MP3 file (step 908). In a first circumstance when the MP3 player needs to play the MP3 file from the middle of the MP3 file, if the totalizer A is greater than or equal to the main\_data\_begin 731 of the current frame 7 (step 910), it is guaranteed that enough data has already been read and stored in the input buffer 102 to correctly process and decode the current frame 7. Therefore, the variable W is changed to be true (for example, a non-zero value), signifying that the bit stream may be correctly processed from then on (step 913).

**Please replace Paragraph [0026] with the following amended paragraph:**

[0026] In a second circumstance when the MP3 player 10 needs to play the MP3 file from the beginning of the MP3 file, since the normal encoding ensures the main\_data\_begin of the first frame being zero, we can use this as an error detection criteria (step 909). In other words, if the main\_data\_begin of the first frame is not zero, the decoder discards this frame and search for the synchronization word in the next frame (step 902). The decoder 103 repeats this process until it finds the frames with a main\_data\_begin having a value of zero.